WHAT IS CLAIMED IS:

1. A system for determining an operating system of a target computer operably connected to a network, the system comprising:

first and second data packets, said first and second data packets compliant with a protocol supported by said network, said first and second data packets transmitted via said network to said target computer;

first and second operating system fingerprints comprising data bits stored in a computer-readable medium, said first and second operating system fingerprints associated with a first operating system;

a first target computer fingerprint comprising data bits stored in a computer-readable medium, said first target computer fingerprint including a representation of at least a portion of data received in response to said transmission of said first data packet;

a second target computer fingerprint comprising data bits stored in a computer-readable medium, said second target computer fingerprint including a representation of at least a portion of data received in response to said transmission of said second data packet; and

fingerprint comparison instructions executable by a computer to compare said first operating system fingerprint and said first target computer fingerprint, to compare said second operating system fingerprint and said second target computer fingerprint, and to generate a result indicative of whether said first operating system was running on said target computer.

2. The system as described in Claim 1, wherein a first range of bits of said first data packet represents a first parameter value, and wherein said first range of bits of said second data packet represents a second parameter value different from said first parameter value.

3. The system as described in Claim 2, wherein said second parameter value is derived by changing one bit in said first range of bits of said first data packet.

4. The system as described in Claim 2, wherein said first and second operating system fingerprints differ.

5. The system as described in Claim 4, further comprising:

a third data packet, said third data packet compliant with said protocol, said first range of bits of said third data packet representing a third parameter value different from said first and second parameter values, said third data packet transmitted via said network to said target computer;

a third operating system fingerprint comprising data bits stored in a computer-readable medium, said third operating system fingerprint associated with said first operating system, said third operating system fingerprint differing from said first and second operating system fingerprints; and

a third target computer fingerprint comprising data bits stored in a computer-readable medium, said third target computer fingerprint including a representation of at least a portion of data received in response to said transmission of said first data packet, said comparison instructions executable by a computer to compare said third operating system fingerprint and said third target computer fingerprint before generating said result.

6. The system as described in Claim 5, further comprising:

fourth, fifth and sixth operating system fingerprints comprising data bits stored in a computer-readable medium, said fourth, fifth and sixth operating system fingerprints associated with a second operating system, at least one of said fourth, fifth and sixth operating system fingerprints differing from a respective one of said first, second and third operating system fingerprints; said comparison instructions executable by a computer to compare said fourth operating system fingerprint and said first target computer fingerprint, to compare said fifth operating system fingerprint and said second target computer fingerprint, to compare said sixth operating system fingerprint and said third target computer fingerprint, and to generate a second result indicative of whether said second operating system was running on said target computer.

7. The system as described in Claim 5, wherein said protocol is TCP/IP and wherein said first range of bits corresponds to a packet field representing a maximum segment size.

8. The system as described in Claim 5, wherein said first parameter value is obtained by setting no bits, said second parameter value is obtained by setting one bit, and said third parameter value is obtained by setting two bits.

9. The system as described in Claim 5, wherein said first parameter value is 0, said second parameter value is 128, and said third parameter value is 128 plus a multiple of 256.

10. The system as described in Claim 5, wherein said first range of bits represents at least two bytes, and wherein a value of said second parameter is obtained by setting the last bit in a byte, and a value for said third parameter is obtained by setting the last bit in a byte.

11. The system as described in Claim 10, wherein said third parameter is obtained by setting adjacent bits in said first range of bits.

12. The system as described in Claim 5, wherein said first, second and third data packets are transmitted in order of lowest parameter value first.

13. A system for determining an operating system of a target computer accessible via a network, the system comprising:

a plurality of data packets compliant with a protocol supported by said network, said plurality of data packets transmitted via said network to said target computer;

a first plurality of operating system fingerprints, each comprising data bits stored in a computer-readable medium, each associated with a first operating system;

a plurality of target computer fingerprints, each comprising data bits stored in a computer-readable medium, each including a representation of at least a portion of data received in response to said transmission of said plurality of data packets; and

fingerprint comparison instructions executable by a computer to compare said first plurality of said operating system fingerprint and said plurality of said target computer fingerprints, and to generate a result indicative of whether said first operating system was running on said target computer.

14. The system as described in Claim 13, wherein said protocol is TCP/IP and wherein each of said plurality of data packets has a different value represented in a respective packet field.

15. The system as described in Claim 14, wherein said packet field is a maximum segment size field.

16. The system as described in Claim 13, further comprising:

a second plurality of operating system fingerprints, each comprising data bits stored in a computer-readable medium, each associated with a second operating system, said fingerprint comparison instructions comparing said second plurality of said operating system fingerprints and said plurality of said target computer fingerprints to generate a second result indicative of whether said second operating system was running on said target computer.

17. A method for determining an operating system of a target computer accessible via a network, the method comprising the steps of:

transmitting to said target computer a plurality of data packets compliant with a protocol supported by said network;

generating a plurality of target computer fingerprints, each including at least a portion of data received via said network in response to said transmission of said plurality of data packets;

comparing said plurality of target computer fingerprints to a first set of predetermined operating system fingerprints, each of said first set of predetermined operating system fingerprints associated with a first operating system; and

generating a result indicative of whether said first operating system was running on said target computer.

18. The method as described in Claim 17, comprising the further steps of:

comparing said plurality of target computer fingerprints to a second set of predetermined operating system fingerprints, each of said second set of predetermined operating system fingerprints associated with a second operating system; and

generating a result indicative of whether said second operating system was running on said target computer.

19. The method as described in Claim 17, wherein said protocol is TCP/IP and wherein some of said plurality of data packets have different values in the same packet field.

20. The method as described in Claim 17, wherein said protocol is TCP/IP and wherein the value of the MSS option of two of said plurality of data packets is divisible by 128.

21. The method as described in Claim 17, wherein a first of said plurality of data packets has a maximum segment size option of 0, wherein a second of said plurality of data packets has a maximum segment size option of 128, and wherein a third of said plurality of data packets has a maximum segment size option of 384.

22. A method for identifying an operating system of a target computer via a network, the method comprising the steps of:

sending a first data packet to said target computer via said network, said first data packet complying with a protocol of said network and having a first pattern of bits in a first range of bits;

generating a first response value representing at least a portion of data received via said network in response to said sending of said first data packet;

sending a second data packet to said target computer via said network, said second data packet complying with said protocol and having a second pattern of bits in a first range of bits, said second pattern of bits different from said first pattern;

generating a second response value representing at least a portion of data received via said network in response to said sending of said second data packet;

sending a third data packet to said target computer via said network, said third data packet complying with said protocol and having a third pattern of bits in a first range of bits, said third pattern of bits different from said first or said second pattern;

generating a third response value representing at least a portion of data received via said network in response to said sending of said third data packet;

comparing said first response value to a first predetermined value associated with a first operating system;

comparing said second response value to a second predetermined value associated with said first operating system;

comparing said third response value to a third predetermined value associated with said first operating system; and

generating a value indicative of a relationship between said first operating system and said target computer.

23. The method as described in Claim 22, the method comprising the further steps of:

comparing said first response value to a fourth predetermined value associated with a second operating system;

comparing said second response value to a fifth predetermined value associated with said second operating system; and

comparing said third response value to a sixth predetermined value associated with said second operating system.

24. The method as described in Claim 22, wherein no bit is set in said first pattern of bits, wherein one bit is set in said second pattern of bits, and wherein two bits are set in said third pattern of bits.

25. The method as described in Claim 22, wherein the number of bytes in said second pattern of bits that have at least one bit set is greater than the number of bytes in said first pattern of bits that have at least one bit set, and wherein the number of bytes in said third pattern of bits that have at least one bit set is greater than the number of bytes in said second pattern of bits that have at least one bit set.

26. The method as described in Claim 22, wherein no byte in said first pattern of bits has a least significant bit or a most significant bit that is set, wherein at least one byte in said second pattern of bits has a most significant bit that is set, and wherein at least one byte in said third pattern of bits has a least significant bit that is set.

27. A system for determining whether a target computer is on a network, the system comprising:

a first set of port identifiers stored in a computer-readable medium, each of said first set of port identifiers representing a port used by computers to receive data packets compliant with a first protocol of said network, each of said first set of port identifiers representing a port associated with known network services;

a first set of data packets, each directed to a port represented by at least one of said first set of port identifiers, each of said first set of data packets compliant with said first protocol and transmitted to said target computer via said network;

a first set of acknowledgement packets received via said network in response to said transmission of said first set of data packets; and

a list of host identifiers, each host identifier representing a computer on said network that transmits data in response to a packet sent to said respective computer, a host identifier representing said target computer added to said list of host identifiers if said first set of acknowledgment packets indicates a responsiveness of said target computer.

28. The system as described in Claim 27, the system further comprising:

a second set of port identifiers stored in a computer-readable medium, each of said second set of port identifiers representing a port used by computers to receive data packets compliant with a second protocol of said network, each of said second set of port identifiers representing a port associated with known network services;

a second set of data packets, each directed to a port represented by at least one of said second set of port identifiers, each of said second set of data packets compliant with said second protocol and transmitted to said target computer via said network, at least one of said second set of data packets including data associated with said known network services;

a second set of acknowledgement packets received via said network in response to said transmission of said second set of data packets; and

a host identifier representing said target computer added to said list of host identifiers if said second set of acknowledgment packets indicates a responsiveness of said target computer.

29. The system as described in Claim 28, wherein said first protocol is TCP, wherein said second protocol is UDP, wherein said second set of acknowledgment packets is a nonzero set of UDP data response packets.

30. The system as described in Claim 27, the system further comprising:

a second set of port identifiers stored in a computer-readable medium, each of said second set of port identifiers representing a port used by computers to receive data packets compliant with a second protocol of said network, each of said second set of port identifiers representing a port associated with known network services;

a second set of data packets, each directed to a port represented by at least one of said second set of port identifiers, each of said second set of data packets compliant with said second protocol and transmitted to said target computer via said network, at least one of said second set of data packets including data associated with said known network services;

a second set of acknowledgement packets received via said network in response to said transmission of said second set of data packets; and

a host identifier representing said target computer added to a second list of host identifiers if said second set of acknowledgment packets does not indicate an unresponsiveness of said target computer, each of said second list of host identifiers representing a computer not known to be unresponsive.

31. The system as described in Claim 30, wherein said first protocol is TCP, wherein said second protocol is UDP, wherein said second set of acknowledgment packets is an empty set of ICMP error packets.

32. The system as described in Claim 30, the system further comprising:

a third set of data packets, each directed to a port represented by at least one of said second set of port identifiers, each compliant with said second

protocol, said third set of data packets transmitted to said target computer throughout a predetermined maximum latency period;

a first response received first in time in response to said transmission of said third set of data packets; and

a second response received second in time in response to said transmission of said third set of data packets, a time duration between said receipt of said first response and said receipt of said second response defining a target computer latency period.

33. The system as described in Claim 32, wherein each of said second set of data packets is transmitted continuously to said target computer for the duration of said target computer latency period.

34. The system as described in Claim 28, the system further comprising:

a third set of data packets, each directed to a port represented by at least one of said second set of port identifiers, each compliant with said second protocol, said third set of data packets transmitted to said target computer throughout a predetermined maximum latency period;

a first response received first in time in response to said transmission of said third set of data packets; and

a second response received second in time in response to said transmission of said third set of data packets, a time duration between said receipt of said first response and said receipt of said second response defining a target computer latency period.

35. The system as described in Claim 34, wherein each of said second set of data packets is transmitted continuously to said target computer for the duration of said target computer latency period.

36. A system for testing the accessibility of a target computer via a network, the system comprising:

a set of port identifiers stored in a computer-readable medium, each of said set of port identifiers representing a UDP-compliant port, at least one of said port identifiers representing a port associated with known network services;

a set of UDP-compliant data packets, each associated with a port represented by at least one of said set of port identifiers, each of said UDP-compliant data packets transmitted continuously to said target computer for a duration approximately the same as the latency period of said target computer, at least one of said UDP-compliant data packets including data associated with said known network services;

a first list representing computers accessible via said network, said first list including said target computer if a nonzero set of UDP data response packets is received in response to said transmission of said data packets; and

a second list representing computers not known to be inaccessible via said network, said second list including said target computer if an empty set of ICMP error packets is received in response to said transmission of said data packets.

37. A method for determining whether a target computer is accessible via a network, the method comprising the steps of:

identifying TCP ports;

sending first data packets to said TCP ports of said target computer, each of said first data packets compliant with TCP;

receiving first acknowledgment packets in response to said sending of said first data packets; and

adding a representation of said target computer to a list representing accessible computers if said first acknowledgment packets are nonzero.

38. The method as described in Claim 37, the method comprising the further steps of:

identifying UDP ports associated with network services;

sending second data packets to said UDP ports of said target computer, at least one of said second data packets sent continuously to said target computer throughout a latency period of said target computer;

receiving second acknowledgment packets in response to said sending of said second data packets; and

adding a representation of said target computer to a list representing accessible computers if said second acknowledgment packets are nonzero UDP data response packets.

39. The method as described in Claim 38, the method comprising the further step of:

determining said latency period of said target computer by measuring the time between responses received in response to packets transmitted to said target computer.

40. The method as described in Claim 38, the method comprising the further step of:

adding a representation of said target computer to a list representing computers not known to be inaccessible via said network, said adding performed if said second acknowledgment packets comprise an empty set of ICMP error packets.

41. A method for assessing the vulnerability of a target computer via a network, the method comprising the steps of:

discovering a set of responsive computers on a network by transmitting a set of ICMP packets, a set of TCP packets and a set of UDP packets to a group of computers on a network;

detecting services on each of said set of responsive computers by transmitting TCP packets to first ports of each of said set of responsive computers and by transmitting UDP packets to second ports of each of said set of responsive computers, said first and second ports commonly used by computers to receive data packets over a network, said TCP packets including data associated with at least one computer-based service known to use one of said first ports, said UDP packets including data associated with at least one computer-based service known to use one of said second ports; and

generating a list of responsive ports using responses received in response to said transmission of said TCP packets and said UDP packets.

42. The method described in Claim 41, the method comprising the further step of:

determining an operating system used by each of said set of responsive computers by comparing predetermined values with portions of responses received from each of said set of responsive computers in response to transmission of a plurality of TCP-compliant packets to each of said set of responsive computers.

43. The method described in Claim 42, the method comprising the further step of:

confirming the presence of vulnerabilities on said network by applying an automated vulnerability script to each responsive port represented in said list of responsive ports, each of said automated vulnerability scripts testing a vulnerability known to be associated with a computer configuration comprising a particular responsive port and a particular operating system.

44. The method described in Claim 43, the method comprising the further step of:

calculating an objective indicia of security of said network, said calculation based on a weighted summation of confirmed vulnerabilities.

45. The method described in Claim 44, the method comprising the further step of:

determining a topology of said network, said topology determination made by transmitting a set of ICMP packets with varying time to live (TTL) settings and by transmitting a set of TCP packets with varying TTL settings.

46. The method described in Claim 45, the method comprising the further step of:

producing a graphical representation of said network, said representation including a topological map of said network, a color-based representation of weighted confirmed vulnerabilities, and an association between said graphical representation and information descriptive of confirmed vulnerabilities and computers on said network.

47. A method of creating a graphical representation of a network, the method comprising the steps of:

obtaining IP addresses of nodes on a network;

obtaining node distance and connectivity relationships between said nodes;

identifying some nodes as routers;

identifying other nodes as leaf nodes connected to one of said routers;

generating graphical representations of router nodes;

for each router, generating graphical representations of directly connected leaf nodes by depicting graphical representations of said directly connected leaf nodes having a spatial relationship to said graphical representation of said respective router; and

depicting links between routers having no intervening routers.

48. The method described in Claim 47, comprising the further step of:

comparing, for each router, the IP address of said respective router and the IP address of each of said directly connected leaf nodes to resolve instances where a directly connected leaf node and said respective router represent two network connections of the same node.

49. A method for creating a topological representation of a network, said method comprising the steps of:

identifying responsive computers on said network;

obtaining a plurality of sequences of IP addresses by sending to each responsive computer a sequence of packets having increasing time to live (TTL) values, each sequence of IP addresses representing nodes in said network between a source computer and one of said responsive computers, adjacent IP addresses in each sequence representing connected nodes, each of said nodes comprising a computer or a router;

generating a list of node structures, each of said node structures including data representing a node and data indicative of other nodes to which it directly connects, said list representing all IP addresses in said plurality of sequences;

determining for each IP address a distance count, said distance count representing a number of nodes between a node having said IP address and a source node;

creating a router structure for each node structure that represents a node comprising a router;

associating with each of said router structures connection data representative of each connecting node that connects to no other node except the router represented by said respective router structure;

for each router structure, visually depicting a graphical shape spatially related to one or more graphical shapes corresponding to connecting nodes represented by said connection data of said respective router structure; and

for each router structure, visually depicting a connection between a graphical shape associated with the respective router structure and another graphical shape associated with a different router structure when distance counts associated with the IP addresses of routers represented by said respective router structure and said different router structure indicate a direct connection.

50. The method described in Claim 49, said method comprising the further step of:

testing whether a router represented by a router structure and a connecting node represented in connection data comprise two network connections of one node.

51. The method described in Claim 49, wherein the graphical shape representing a router is a sphere, and wherein each of said spatially related graphical shapes is a sphere orbiting said sphere representing said router.

52. A method for calculating an objective vulnerability score, said method comprising the steps of:

identifying known vulnerabilities of a network;

weighting said known vulnerabilities based on either ease of exploitation or level of access granted; and

determining a vulnerability value numerically representing a combination of weighted known vulnerabilities of a network.

53. A method for calculating an objective security score for a network, said method comprising the steps of:

determining a vulnerability value numerically representing a combination of known vulnerabilities of a network;

determining an exposure value numerically representing a combination of accessible ports of computers on said network; and

deriving a score by combining said vulnerability value and said exposure value.

54. The method described in Claim 53, wherein said combination of known vulnerabilities is a summation of weighted numeric expressions of particular vulnerabilities, the weighting based on an ease of exploitation ranking and on an access granted ranking for each vulnerability.

55. A method for conducting an automated network vulnerability attack, said method comprising the steps of:

selecting a set of vulnerability attacks for each responsive computer on a network, each selected vulnerability attack for each responsive computer designed to expose a vulnerability associated with ports of said respective computer known to be accessible and also associated with an operating system used by said respective computer;

encoding said set of vulnerability attacks such that each is represented in a database by a unique identifier;

representing each of said set of vulnerability attacks using instructions of an automated scripting language; and

executing said vulnerability attacks by processing said instructions with a computer.

56. A hierarchical network vulnerability report, comprising:

a first report level comprising:

an objective score representing the security of said network; and

a graphical representation of a network topology, including a graphical representation of computers accessible via said network and a color-based graphical representation of the vulnerability of at least some of said computers;

and

a second report level comprising:

a textual list describing said computers and their associated vulnerabilities; and

an exposure report describing accessible ports and services of said computers.

57. A vulnerability assessment language comprising:

a set of programming language statements used to create executable scripts, said scripts executed in a thread-safe execution architecture wherein all variables are stack variables and wherein a parse tree is treated as a read-only data structure;

a set of special scalar data types interchangeable with an integer data type in expressions, each of said set of special scalar data types having a set of constant values configured to support vulnerability assessment operations embodied in scripts;

a set of native objects declared in a metascope owning a script scope to make available said native objects to executable scripts, said native objects facilitating network communication, providing callable member functions for building lists of unique ports and directing script execution to certain hosts, and providing IP addresses for scripts; and

a vulnerability object behaving to copy itself into a global data area where other scripts may access its information to compromise another machine, facilitating the use by one script of vulnerability data discovered by a different script.

58. A method for automated application of a known vulnerability on a target computer, the method comprising the steps of:

providing a database of known vulnerabilities, the database including a data object;

providing an executable script, the executable script associated with the data object;

applying the executable script to the target computer, the script performing the known vulnerability on a port of the target computer; and

returning a value representing at least one of the success, failure or other outcome of the executable script.

59. A method for automated application of known vulnerabilities to target computers of a network, the method comprising the steps of:

providing a database of known vulnerabilities;

providing a set of executable scripts, each executable to apply a known vulnerability to a specified target computer;

executing first executable scripts to apply vulnerabilities on specified target computers;

monitoring return values representing a success, failure or other outcome of each of said first executable scripts; and

generating a report using said return values, said report representing a security level of said network.

60. The method described in Claim 59, comprising the further step of:

identifying execution time intervals wherein execution of said first executable scripts commences at the beginning of each of said time intervals and pauses at the end of each of said time intervals, until all of said first executable scripts have executed.

61. The method described in Claim 60, comprising the further step of:

automatically repeating said execution of said first executable scripts when said execution of said first executable scripts is completed.

62. The method described in Claim 61, comprising the further steps of:

generating a report upon each completed execution of said first executable scripts; and

calculating a security trend for said network by comparing a plurality of said reports.

63. The method described in Claim 59, comprising the further step of:

executing second executable scripts to apply vulnerabilities to a second network of computers during said execution of said first executable scripts.

64. The method described in Claim 63, wherein said second network is a subset of said network.

65. The method described in Claim 59, wherein said first executable scripts are configured to apply vulnerabilities to a first port of all of said target computers before applying vulnerabilities to a second port of all of said target computers.

66. The method described in Claim 57, the method comprising the further step of:

allocating a plurality of packet slots, each packet slot permitting asynchronous transmission of a packet by one of said executable scripts.